

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A computer program product, tangibly embodied in a machine-readable storage device, the computer program product being operable to cause data processing apparatus to perform operations comprising:

receiving run-time code for an application, the run-time code being generated from a converted design-time representation of the application, the run-time code comprising a flag indicating that the run-time code was generated from the converted design-time representation, wherein:

the converted design-time representation of the application is generated from an original design-time representation of the application developed for use in a first run-time environment for executing applications having been developed in a first design-time environment, the converted design-time representation is stored as metadata in a metadata repository, the converted design-time representation having been developed by a development tool based on a metamodel that defines metamodel objects, the first design-time environment using a first programming model comprising one or more first model elements including screens and processing logic for each screen, the original design-time representation including one or more application screens and original processing logic for each application screen, the original processing logic including a call to a run-time module in the first run-time environment; and

the converted design-time representation of the application is for use in a second run-time environment for executing applications having been developed in a second design-time environment, the second design-time environment using a second programming model comprising one or more second model elements including models, views, and controllers, the converted design-time representation of the application including one or more application views based on the one or more application screens, and converted processing logic based on the original processing logic, the converted processing logic being a design-time representation used to generate run-time code for the original processing logic, the run-time code for the original processing logic is configured to be being capable of being executed in the second run-time environment; and

executing the run-time code in the second run-time environment using an adapter operable to interface with the run-time module in the first run-time environment, comprising using the adapter to perform a function comprising input validation not performed by the original processing logic.

2. (Original) The computer program product of claim 1, wherein the first programming model is the SAP Dynpro programming model and the second programming model is the SAP Web Dynpro programming model.

3. (Cancelled).

4. (Cancelled).

5. (Previously Presented) The computer program product of claim 1, wherein the function comprises input formatting.
6. (Original) The computer program product of claim 1, wherein:
 - the original design-time representation of the application comprises original state control logic; and
 - the converted design-time representation of the application comprises converted state control logic based on the original state control logic, the converted state control logic capable of being executed by the adapter.
7. (Original) The computer program product of claim 1, wherein:
 - the original design-time representation of the application comprises one or more controls from a first set of controls;
 - the converted design-time representation of the application comprises one or more controls from a second set of controls, each control in the converted design-time representation of the application corresponding to a control in the original design-time representation of the application; and
 - executing the run-time code comprises rendering the controls in the converted design-time representation of the application.
8. (Currently Amended) A computer program product, tangibly embodied in a machine-readable storage device, the computer program product being operable to cause data processing apparatus to perform operations comprising:

receiving run-time code for an application, the run-time code comprising a flag indicating whether the run-time code was generated from a native design-time representation or a converted design-time representation;

determining whether the run-time code was generated from the native design-time representation of the application or from the converted design-time representation of the application based on the flag, wherein:

the native design-time representation of the application is for use in a first run-time environment for executing applications having been developed in a first design-time environment, the first design-time environment using a first programming model comprising one or more first model elements including models, views, and controllers; and

the converted design-time representation of the application is generated from an original design-time representation of the application developed for use in a second run-time environment for executing applications having been developed in a second design-time environment, the converted design-time representation is stored as metadata in a metadata repository, the converted design-time representation having been developed by a development tool based on a metamodel that defines metamodel objects, the second design-time environment using a second programming model comprising one or more second model elements including screens and processing logic for each screen, the original design-time representation including one or more application screens and original processing logic for each application screen, the converted design-time representation of the application including one or more application views based on the one or more application screens, and converted

processing logic based on the original processing logic, the converted processing logic
being a design-time representation used to generate run-time code for the original
processing logic, the run-time code for the original processing logic is configured to be
being capable of being executed in the second run-time environment;

if the run-time code was generated from the native design-time representation,
executing the run-time code in the first run-time environment using a set of run-time
modules in the first run-time environment; and

if the run-time code was generated from the converted design-time
representation, executing the run-time code in the first run-time environment using an
adapter operable to interface with a set of run-time modules in the second run-time
environment, comprising using the adapter to perform a function including input
validation not performed by the original processing logic.

9. (Original) The computer program product of claim 8, wherein the first
programming model is the SAP Web Dynpro programming model and the second
programming model is the SAP Dynpro programming model.

10. (Cancelled).

11. (Original) The computer program product of claim 8, wherein:
executing the run-time code using the set of run-time modules in the first run-time
environment comprises using a first sequence of process steps; and

executing the run-time code using the set of run-time modules in the second run-time environment comprises using a second sequence of process steps.

12. (Currently Amended) An apparatus comprising:

means for receiving run-time code for an application, the run-time code being generated from a converted design-time representation of the application, the run-time code comprising a flag indicating that the run-time code was generated from the converted design-time representation, wherein:

the converted design-time representation of the application is generated from an original design-time representation of the application developed for use in a first run-time environment for executing applications having been developed in a first design-time environment, the converted design-time representation is stored as metadata in a metadata repository, the converted design-time representation having been developed by a development tool based on a metamodel that defines metamodel objects, the first design-time environment using a first programming model comprising one or more first model elements including screens and processing logic for each screen, the original design-time representation including one or more application screens and original processing logic for each application screen, the original processing logic including a call to a run-time module in the first run-time environment; and

the converted design-time representation of the application is for use in a second run-time environment for executing applications having been developed in a second design-time environment, the second design-time environment using a second

programming model comprising one or more second model elements including models, views, and controllers, the converted design-time representation of the application including one or more application views based on the one or more application screens, and converted processing logic based on the original processing logic, the converted processing logic being a design-time representation used to generate run-time code for the original processing logic, the run-time code for the original processing logic is configured to be being capable of being executed in the second run-time environment; and

means for executing the run-time code in the second run-time environment using an adapter operable to interface with the run-time module in the first run-time environment, comprising means for using the adapter to perform a function including input validation not performed by the original processing logic.

13. (Original) The apparatus of claim 12, wherein the first programming model is the SAP Dynpro programming model and the second programming model is the SAP Web Dynpro programming model.

14. (Original) The apparatus of claim 12, wherein:

the original design-time representation of the application comprises original state control logic; and

the converted design-time representation of the application comprises converted state control logic based on the original state control logic, the converted state control logic capable of being executed by the adapter.

15. (Cancelled).

16. (Currently Amended) A method comprising:

receiving run-time code for an application, the run-time code being generated from a converted design-time representation of the application, the run-time code comprising a flag indicating that the run-time code was generated from the converted design-time representation, wherein:

the converted design-time representation of the application is generated from an original design-time representation of the application developed for use in a first run-time environment for executing applications having been developed in a first design-time environment, the converted design-time representation is stored as metadata in a metadata repository, the converted design-time representation having been developed by a development tool based on a metamodel that defines metamodel objects, the first design-time environment using a first programming model comprising one or more first model elements including screens and processing logic for each screen, the original design-time representation including one or more application screens and original processing logic for each application screen, the original processing logic including a call to a run-time module in the first run-time environment; and

the converted design-time representation of the application is for use in a second run-time environment for executing applications having been developed in a second design-time environment, the second design-time environment using a second programming model comprising one or more second model elements including models,

views, and controllers, the converted design-time representation of the application including one or more application views based on the one or more application screens, and converted processing logic based on the original processing logic, the converted processing logic being a design-time representation used to generate run-time code for the original processing logic, the run-time code for the original processing logic is configured to be ~~being capable of~~ being executed in the second run-time environment; and

executing the run-time code in the second run-time environment using an adapter operable to interface with the run-time module in the first run-time environment, comprising using the adapter to perform a function including input validation not performed by the original processing logic.

17. (Original) The method of claim 16, wherein the first programming model is the SAP Dynpro programming model and the second programming model is the SAP Web Dynpro programming model.

18. (Original) The method of claim 16, wherein:
the original design-time representation of the application comprises original state control logic; and
the converted design-time representation of the application comprises converted state control logic based on the original state control logic, the converted state control logic capable of being executed by the adapter.

19. (Cancelled).